International Journal of Electrical and Computer Engineering (IJECE)
Vol. 5, No. 3, June 2015, pp. 579~585
ISSN: 2088-8708

Fingerprint Authentication Schemesfor Mobile Devices

Jinho Han
Department of Liberal Studies (Computer), Korean Bible University, Korea

Article Info

Article history:

Received Jan 17, 2015
Revised Apr 20, 2015
Accepted May 5, 2015

Keyword:

Biometrics
Cryptography
Fingerprint Templates
Mobile Devices

ABSTRACT

In certain applications, fingerprint authentication systems require templates to be stored in databases. The possible leakage of these fingerprint templates can lead to serious security and privacy threats. Therefore, with the frequent use of fingerprint authentication on mobile devices, it has become increasingly important to keep fingerprint data safe. Due to rapid developments in optical equipment, biometric systems are now able to gain the same biometric images repeatedly, so strong features can be selected with precision. Strong features refer to high-quality features which can be easily distinguished from other features in biometric raw images. In this paper, we introduce an algorithm that identifies these strong features with certain probability from a given fingerprint image. Once values are extracted from these features, they are used as the authentication data. Using the geometric information of these strong features, a cancelable fingerprint template can be produced, and the process of extracting values and geometric information is further explained. Because this is a light-weight authentication scheme, this template has practical usage for low performance mobile devices. Finally, we demonstrate that our proposed schemes are secure and that the user's biometric raw data of the fingerprint are safe, even when the mobile device is lost or stolen.

Corresponding Author:

Jinho Han,
Department of Liberal Studies (Computer),
Korean Bible University,
32 Dongil-ro(st) 214-gil, Nowon-gu, Seoul, Korea,
Email: hjinob@bible.ac.kr

1. INTRODUCTION

The fingerprint authentication system is now widely considered to be a convenient human identification method for mobile devices [1]. In the near future, one mobile device will have one or more biometric templates such as fingerprint templates and this calls for higher security measures to keep the data safe. A fingerprint authentication system consists of an application and its template stored in a database. The leakage of these fingerprint templates entails serious security and privacy threats.

In this paper, we propose secure and efficient fingerprint authentication schemes which are designed to suit mobile devices, and explain the process of producing cancelable fingerprint templates that protect the user's privacy. Our suggestion is based on the assumption that the fingerprint samples are of robust quality, neither partial nor wet, with good condition images. Since the user has enough time to authenticate his own fingerprint through multiple trials, this assumption is reasonable. Strong features from the fingerprints are used for this scheme. Strong features refer to high-quality features which can be distinguished easily from other features in biometric raw images [2]. Through our *(w-k) Select* algorithm we will show that strong features can be found with certain probability. We extract values from the features and their geometric information. The proposed cancelable fingerprint template is composed of these values and geometric information extracted from these features. The proposed schemes are produced using this template.

Geometric hashing [3] is a method for finding two-dimensional objects. We use this method to extract coordinates *(x, y)* values of the strong features.

Section 2 contains a discussion of previous work related to secure authentication systems using biometrics. Section 3 describes the process of selecting strong features from fingerprint images and generating cancelable fingerprint templates. In section 4, we then propose our schemes in which the hash values of features are used as verification information by the authentication system. Because our proposed schemes are light-weight and efficient, they suit low performance mobile devices. In section 5, we demonstrate that our proposed schemes based on one-time password (OTP) are secure, and the user's biometric data of fingerprints are also safe, even when the mobile device is lost or stolen. Finally, conclusions are discussed in section 6.

2. RELATED WORK

Since the introduction of outstanding concepts such as fuzzy commitment [4] and fuzzy vault [5] schemes, which locks biometric data for safety, there have been studies on how to protect biometric templates by using robust hash functions [6, 7]. To generate cancelable fingerprint templates Ratha et al. proposed a hard-to-invert transformation [8, 9]. By carefully selecting strong features that are easier for a specific user to replicate, Randomized Biometric Templates (RBTs) [2] have also been proposed, creating a difficult environment for attackers to make guesses. Biometric key extraction is a method to get fixed binary from biometric templates. Many studies have also suggested methods of making cryptographic keys from various biometrics. Iris [10, 11], face [12] and finger vein [13] are some examples.

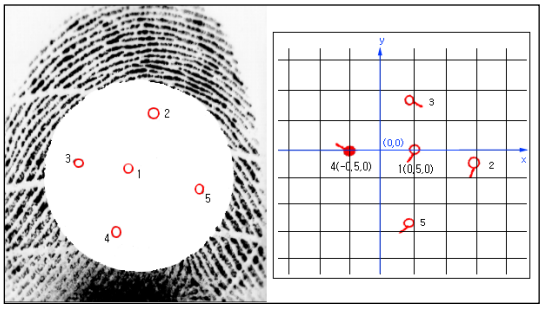

Figure 1. Selecting strong features and geometric hashing of the features

Algorithm 1. Specification of the *(w-k)Select* algorithm

Input: fingerprint samples $(\beta_1,\ldots,\beta_w)$, w, k
Output: the strong features F
1: $k_i = 0$ $(i=1,\ldots,N)$ // N is the number of all featuresof original fingerprint β
2: **for** $j \leftarrow 1$ **to** w **do** {
3: $(f_1,\ldots,f_N) \leftarrow$ Find(β_j) //Find fingerprint features
4: **for** $i \leftarrow 1$ **to** N **do** {
5: **if** f_i exists **then** k_i++
6: }
7: }
8: $F \leftarrow f_i$ has k_i such that $k_i \geq k$
9: **return** F

3. GENERATION OF TEMPLATE

3.1. Strong Features of Fingerprint

It is widely known that biometric data cannot be precisely reproduced each time it is measured. When a biometric system uses these data, it has to solve the problem of error tolerance. However, due to the development of optical equipment, the biometric systems are now able to repeatedly obtain almost the same biometric images. Strong features are high-quality features which can be distinguished easily from other features in biometric raw images. Using an error correction method like quantization [2], we can arrive at the same and repeatable values from the strong features. An example of selecting five strong features in a

fingerprint image is shown in Figure 1. Strong features of certain probability are results of the proposed *(w-k)Select* algorithm which is shown in Algorithm 1. If some strong features of 100 percent probability are found, they can be used as cryptographic keys. Here, w is the number of input fingerprint images and k is the minimum number of trials to find the features. We say F is the set of *(w-k)* strong features. When ϕ = *(k/w)**100, we can say F is the set of *(ϕ%)* strong features.

3.2. Generating Fingerprint Template with Strong Features

After enrollment pre-processing phase which consists of one-pixel-width thinning stage and positioning stage with an enrolled fingerprint image, we can select features, called as minutiae, as shown in Figure 1 and extract valid minutiae values. A minutia can be specified by its coordinates *(x, y)*, angle *(θ)*, and its type *(t)* which is ending or bifurcation, such as $m_i = (x_i, y_i, \theta_i, t_i)$ [14]. To apply error correction to minutiae m_i, each minutia should be quantized to the ranges of values, where x_i and y_i may have 16 ranges (0-15), θ_i 32 ranges (0-31), and t_i 4 ranges (0-3), respectively. An example of geometric hashing with five minutiae is shown in Figure 1. With this method, we can extract minutiae information from strong features. Table 1 shows hash table of vector $\overrightarrow{P_4P_1}$ which has *x, y* coordinates of Figure 1's five minutiae with basis (4,1). If other hash tables for the strong features are needed, hash table of vector $\overrightarrow{P_4P_2}$ or vector $\overrightarrow{P_4P_3}$ can be made, respectively.

The geometric information of five minutiae is shown in Figure 2, which are distances (d_1,d_2,d_3,d_4) and angles (a_1,a_2,a_3,a_4) between strong features (F_1,F_2,F_3,F_4,F_5). Our fingerprint template is composed of geometric information and minutiae values of strong features. In the verification phase, geometric information is provided for the alignment of input fingerprint in searching for strong features. Let n be a number of strong features. Geometric information G is as shown below

$$G = (d_1,\ldots,d_{n-1}, a_1,\ldots,a_{n-1}).$$

Minutiae values are actual authentication data that is used in user verification. Original minutiae values M is as shown below

$$M = (m_1,\ldots,m_n).$$

Finally, our fingerprint template is as follows

$$FingTemp = (G \,||\, M).(\,||\,\text{ : binary concatenation})$$

In order to check security functions, we will randomize minutiae values and insert them into the template. Randomizing minutiae information will be explained in section 4.

Table 1. Hash table of strong features with basis (4,1)

x	y	basis	point
0.5	0	(4,1)	1
1.4	-0.25	(4,1)	2
0.4	0.8	(4,1)	3
-0.5	0	(4,1)	4
0.4	-1.3	(4,1)	5

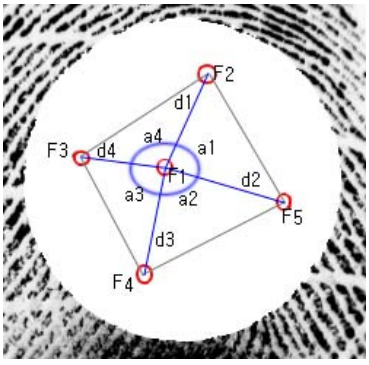

Figure 2. Distances and angles between strong features

4. OUR CONSTRUCTION

We have proposed one basic scheme and two extended schemes. In scheme 1, we simply use *FingTemp* and random number *r*. In scheme 2, we *use FingTemp, r*, and user's password π to increase security and in scheme 3, we consider the case of losing one of *n* strong features. We assume that with error correction such as simple quantization, we can obtain the *(100%)* strong features of the same user in our schemes.

4.1. Scheme 1

As explained above, let $FingTemp = (G \,||\, M)$ be the original fingerprint template given from the raw image.

●**Enrollment:** Given a security parameter *k*, let *p* be a prime number. The enrollment algorithm chooses a random number $r \in Z_p^*$ and hash functions $H_1 : \{0,1\}^* \rightarrow Z_p^*$, and $H_2 : Z_p^* \rightarrow \{0,1\}^k$.

1) Compute *M* of *FingTemp*, $M_{h1}=H_1(M)$.
2) Compute $M' = M_{h1} \oplus r$.($\oplus$: bitwise exclusive-or, XOR)
3) Compute $r_{h2} = H_2(r)$.

Later, r_{h2} will be used as verification information. Randomized minutiae *M'* is taken to make a cancelable fingerprint template. Our cancelable fingerprint template is as follows

$$CanFingTemp = (G \,||\, M').$$

If a user's *ID* is necessary for authentication in the device, *ID*, *CanFingTemp* and r_{h2} are stored in a database as follows

$$\text{DB record: } \{ID, (G \,||\, M'), r_{h2}\}.$$

●**Verification:** With the use of the input biometric template which is obtained by a fingerprint scanner, someone's minutiae values β is provided to the verification algorithm. Geometric information *G* is used to find exact position of the minutiae.

1) Compute $\beta_{h1} = H_1(\beta)$.
2) Compute $\beta' = \beta_{h1} \oplus M'$.
3) Compute $(\beta')_{h2} = H_2(\beta')$.
4) If $(\beta')_{h2} = r_{h2}$ then output "Yes" which indicates that the user is authentic, or "No" which means that the user is not.

4.2. Scheme 2

Let π be the user's password. For a more secure verification process we use the user's fingerprint template and password, that is to say a two-factor authentication.

●**Enrollment:** Given a security parameter *k*, let *p* be a prime number. The enrollment algorithm chooses a random number $r \in Z_p^*$ and hash functions $H_1 : \{0,1\}^* \rightarrow Z_p^*$, and $H_2 : Z_p^* \rightarrow \{0,1\}^k$.

1) Compute *M* of *FingTemp*, $M_{h1}=H_1(M)$
2) Compute $\pi_{h1} = H_1(\pi)$.
3) Compute $M' = M_{h1} \oplus r$.
4) Compute $M'' = M' \oplus \pi_{h1}$
5) Compute $r_{h2} = H_2(r)$.

Later, r_{h2} will be used as verification information. Randomized minutiae *M''* is taken to make a cancelable fingerprint template. Now, our fingerprint template is as follows

$$CanFingTemp = (G \,||\, M'').$$

If the user's *ID* is necessary for authentication in the device, *ID*, *CanFingTemp* and r_{h2} are stored in a database as follows

$$\text{DB record: } \{ID, (G \,||\, M''), r_{h2}\}.$$

●**Verification:** With the use of the input biometric template which is obtained by a fingerprint scanner, someone's minutiae values β and also his password μ are provided to the verification algorithm. Geometric information *G* is used to find the exact position of the minutiae.

1) Compute $\beta_{h1} = H_1(\beta)$.
2) Compute $\mu_{h1} = H_1(\mu)$.
3) Compute $\beta' = \beta_{h1} \oplus M''$.
4) Compute $\beta'' = \beta' \oplus \mu_{h1}$.
5) Compute $(\beta'')_{h2} = H_2(\beta'')$.
6) If $(\beta'')_{h2} = r_{h2}$ then output "Yes" which indicates that the user is authentic, or "No" which means that the user is not.

4.3. Scheme 3

As explained above, let $FingTemp = (G \,||\, M)$ be the original fingerprint template given from the raw image. Here we consider the case of losing one of n strong features. We make $FingTemp_{(-1)}, \ldots, FingTemp_{(-n)}$ such that $FingTemp_{(-i)} = (G \,||\, M_{(-i)})$ and $M_{(-i)} = M - m_i$.

●**Enrollment:** Given a security parameter k, let p be a prime number. The enrollment algorithm chooses random numbers $r \in Z_p^*$ and $(r_{(-1)}, \ldots r_{(-n)}) \in Z_p^*$, two hash functions $H_1 : \{0,1\}^* \rightarrow Z_p^*$, and $H_2 : Z_p^* \rightarrow \{0,1\}^k$.

1) Compute M of $FingTemp$, $M_{h1}=H_1(M)$.
   Compute $M_{(-1)h1}=H_1(M_{(-1)})$.
   …
   Compute $M_{(-n)h1}=H_1(M_{(-n)})$.
2) Compute $M' = M_{h1} \oplus r$.
   Compute $M'_{(-1)} = M_{(-1)h1} \oplus r_{(-1)}$.
   …
   Compute $M'_{(-n)} = M_{(-n)h1} \oplus r_{(-n)}$.
3) Compute $r_{h2} = H_2(r)$.
   Compute $r_{(-1)h2} = H_2(r_{(-1)})$.
   …
   Compute $r_{(-n)h2} = H_2(r_{(-n)})$.

Later, $r_{h2}, r_{(-1)h2}, \ldots, r_{(-n)h2}$ will be used as verification information. Randomized minutiae $M', M'_{(-1)}, \ldots, M'_{(-n)}$ are used in order to make cancelable fingerprint templates. Our cancelable fingerprint templates are as follows

$$CanFingTemp = (G \,||\, M').$$
$$CanFingTemp_{(-1)} = (G \,||\, M'_{(-1)}).$$
$$\ldots$$
$$CanFingTemp_{(-n)} = (G \,||\, M'_{(-n)}).$$

If a user's ID is necessary for authentication in the device, $ID, r_{h2}, r_{(-1)h2}, \ldots, r_{(-n)h2}$, and our cancelable fingerprint templates are stored in a database as follows

DB record: $\{ID, (G \,||\, M'), r_{h2}\}$.
$\{ID, (G \,||\, M'_{(-1)}), r_{(-1)h2}\}$.
…
$\{ID, (G \,||\, M'_{(-n)}), r_{(-n)h2}\}$.

●**Verification:** With the use of the input biometric template which is obtained by a fingerprint scanner, someone's minutiae values β is provided to the verification algorithm. Geometric information G is used to find the exact position of the minutiae. We assume that β has n or $n-1$ minutiae values.

1) Compute $\beta_{h1} = H_1(\beta)$.
2) Compute $\beta' = \beta_{h1} \oplus M'$.
   Compute $\beta'_{(-1)} = \beta_{h1} \oplus M'_{(-1)}$.
   …
   Compute $\beta'_{(-n)} = \beta_{h1} \oplus M'_{(-n)}$.
3) Compute $(\beta')_{h2} = H_2(\beta')$.
   Compute $(\beta'_{(-1)})_{h2} = H_2(\beta'_{(-1)})$
   …
   Compute $(\beta'_{(-n)})_{h2} = H_2(\beta'_{(-n)})$
4) If $(\beta')_{h2} = r_{h2}$ or $(\beta'_{(-1)})_{h2} = r_{(-1)h2}$ or … or $(\beta'_{(-n)})_{h2} = r_{(-n)h2}$ then output "Yes" which indicates that the user is authentic, or "No" which means that the user is not.

4.4. Correctness

In scheme 1 and 3, if $M = \beta$ then the following equation is correct.

$$\begin{aligned}(\beta')_{h2} &= H_2(\beta') \\ &= H_2(\beta_{h1} \oplus M') \\ &= H_2(H_1(\beta) \oplus M') \\ &= H_2(H_1(\beta) \oplus H_1(M) \oplus r) \\ &= H_2(r) \\ &= r_{h2}.\end{aligned}$$

And in scheme 3, if $M_{(-i)} = \beta$ then the following equation is correct.

$$\begin{aligned}(\beta'_{(-i)})_{h2} &= H_2(\beta'_{(-i)}) \\ &= H_2(\beta_{h1} \oplus M'_{(-i)})\end{aligned}$$

$= H_2(H_1(\beta) \oplus M'_{(-i)})$
$= H_2(H_1(\beta) \oplus H_1(M_{(-i)}) \oplus r_{(-i)})$
$= H_2(r_{(-i)})$
$= r_{(-i)h2}$.

Finally, in scheme 2, if $M = \beta$ and $\pi = \mu$ then the following equation is correct.

$(\beta'')_{h2} = H_2(\beta'')$
$= H_2(\beta' \oplus \mu_{h1})$
$= H_2(\beta_{h1} \oplus M'' \oplus \mu_{h1})$
$= H_2(\beta_{h1} \oplus M' \oplus \pi_{h1} \oplus \mu_{h1})$
$= H_2(\beta_{h1} \oplus M' \oplus H_1(\pi) \oplus H_1(\mu))$
$= H_2(\beta_{h1} \oplus M')$
$= H_2(\beta_{h1} \oplus M_{h1} \oplus r)$
$= H_2(H_1(\beta) \oplus H_1(M) \oplus r)$
$= H_2(r)$
$= r_{h2}$.

5. SECURITY ANALYSIS

In this section we perform security analysis of our fingerprint template and our proposed schemes.

●**Security for Fingerprint Template:** In the previous section we have shown the process of selecting strong features for the template from the original fingerprint images as an example. The proposed cancelable fingerprint template, *CanFingTemp* consists of two components, geometric information G and randomized minutiae M' in scheme 1, 3 (or M'' in scheme 2). First, we discuss security for M' (or M''). According to the error-correction quantization of m_i, we assume that original one minutia information may have the size of 15 bits such as $m_i = (x_i, y_i, \theta_i, t_i)$, where at least x_i and y_i may have 16 relative different values (4bits, 0-15), θ_i 32 relative values (5bits, 0-31), and t_i 4 values (2bits, 0-3), respectively. When the number of strong features $n = 5$, M is a 75 bit long value and in the case of $n = 10$, M is a 150 bit long value. The length of M is long enough, which makes it harder for attackers to take a guess. Randomized minutiae M' (or M'') is computed as follows. M is transformed with the one-way hash function ($H_1(\cdot)$) and randomized using random number r with XOR operation. In scheme 2, M is randomized once more using hash value of π with XOR operation. Random number r also is transformed with the hash function ($H_2(\cdot)$) and this hash value r_{h2} is stored in the device. After that, r is deleted from the device, so nobody can compute the original minutiae information M.

Next, note that G is the information of relative distances (d_i) and angles (a_i) between strong features. When G is compromised by an attack, the "old G" can be canceled and a new fingerprint template with "new G" can be enrolled, which means that other strong features will be selected for the new template. If we find 20 strong features from the user's fingerprint and select 10 features from among them, we can make new templates more than 180,000 times (combination of 20 elements choose 10). In short, our *CanFingTemp* is cancelable and it keeps the user's fingerprint data safe.

●**Security for Proposed Schemes:** In our schemes, random number r is transformed with the one-way hash function ($H_2(\cdot)$) and hash value r_{h2} (or $r_{(-i)h2}$) is stored in a database as user's verification information. Random number r (or $r_{(-i)}$) is eliminated from the device and r (or $r_{(-i)}$) cannot be restored from the hash value r_{h2} (or $r_{(-i)h2}$). Each time user's enrollment is implemented, random r (or $r_{(-i)}$) is a different value which is then used to transform minutiae information M differently. In short, random r (or $r_{(-i)}$) is used as a one-time password (OTP) in our schemes which adds a high level of security to the proposed authentication schemes. We can say that our authentication schemes based on one-time passwords are secure.

6. CONCLUSION

In this paper, we propose a cancelable fingerprint template which uses high quality and easily distinguishable features. The proposed algorithm, *(w-k)Select* outputs strong features of certain probability. Because we only use partial features of fingerprints, even when partial information is lost, the user's fingerprint data is safe. Our cancelable template is composed of geometric information G which is the relative value of distances and angles between strong features and randomized minutiae M' of strong features. With this cancelable fingerprint template and random number r (and user's password π), it is evident that our scheme is secure. Our schemes perform only bitwise XOR operations in enrollment and verification phases, which means they are light-weight schemes which are well suited for low performance mobile devices. We believe that our schemes are useful for both the protection of fingerprint data and human authentication on mobile devices.

REFERENCES

[1] Apple Inc. Apple Online Store, "iPhone6 Touch ID, at http://www.apple.com/iphone-6/touch-id/," 2015.
[2] L. Ballard, S. Kamara, F. Monrose, and M. Reiter, "Towards Practical Biometric Key Generation with Randomized Biometric Templates", *CCS'08*, October 27-31, 2008, Alexandria, Virginia, USA, 2008.
[3] H. Wolfson and I. Rigoutsos, "Geometric Hashing: An Overview", *IEEE Computational Science and Engineering*, 4(4), pp. 10-21, 1997.
[4] A. Juels and M. Wattenberg, "A fuzzy commitment scheme", in *Proc. ACM Conf. on Computer and Communications Security*, pp. 28-36, 1999.
[5] A. Juels and M. Sudan, "A fuzzy vault scheme", in *IEEE Intl. Symp. On Information Theory*, 2002.
[6] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmhashing: a novel approach for cancelable biometrics", *Information Processing Letters*, Vol 93, pp. 614-634, 2005.
[7] Y. Sutcu, T. Sencar, and N. Memon, "A secure biometric authentication scheme based on robust hashing", In *ACM MMSEC Workshop*, 2005.
[8] N. Ratha, J. Connell, and R. Bolle, and S. Chikkerur, "Cancelable biometrics: A case study in fingerprints", In *Intl. Conf. on Pattern Recognition*, pp. 370-373, 2006.
[9] N. Ratha, S. Chikkerur, J. Connell, and R. Bolle, "Generating cancelable fingerprint templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 29. No.4, pp. 561-172, 2007.
[10] Y.J. Lee, H.G. Lee, K.R. Park, and J. Kim, "Invariant biometric key extraction based on iris code", *Proc. of IEEK Fall Conf*., Seoul, Korea, vol. 28, no. 2, 2005.
[11] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively", *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1081-1088, 2006.
[12] A. Gohand D.C.L. Ngo, "Computation of cryptographic keys from face biometrics", *International Federation for Information Processing 2003, Springer-Verlag, LNCS 2828*, pp. 1–13, 2003.
[13] A. Venckauskas and P. Nanevicius, "Cryptographic Key Generation from Finger Vein", *International Journal of Engineering Sciences and Research Technology*. 2(4), pp. 733–738, 2013.
[14] D. Moon, *et al*., "Fingerprint Template Protection using Fuzzy Vault", *LNCS 4707*, pp. 1141-1151, 2007.

BIOGRAPHY OF AUTHOR

Jinho Han was born in Seoul, Korea on April 5th, 1965; He received the B.A. degree in the Department of Forestry from Korea University and M.E. degree in the Department of Network Management at Dongguk University, Seoul, Korea, in 1990 and 2006, respectively. He received the Ph.D. degree at the Graduate School of Information Security from Korea University in 2013. He is currently anassistant professor in the department of Liberal studies (computer) at Korean Bible University, Seoul, Korea. His research areas include broadcast encryption, attribute-based encryption, and biometrics.